НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу

«До захисту допущено»
В.О. завідувача кафедри
_____ О.Л. Тимощук

Дипломна робота

на здобуття ступеня бакалавра

з напряму підготовки 6.050101 "Комп'ютерні науки"

на тему: «Методика прогнозування платоспроможності на основі машинного навчання»

Виконав:
студент IV курсу, групи КА-55
Скидан Богдан Олегович _____

Керівник:
доцент кафедри ММСА,
Мілявський Ю. Л. _____

Консультант з економічного розділу:
доцент, к.е.н. Шевчук О. А. _____

Консультант з нормоконтролю:
доцент, к.т.н. Коваленко А.Є. _____

Рецензент:
к.т.н. Мурга М.О _____

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2019 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу

Рівень вищої освіти – перший (бакалаврський)

Напрям підготовки – 6.050101 "Комп'ютерні науки"

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ О.Л. Тимощук
(підпис)

«___»_____2019 р.

ЗАВДАННЯ
на дипломну роботу студенту
Скидану Богдану Олеговичу
(прізвище, ім'я, по батькові)

1. Тема роботи Методика прогнозування платоспроможності на основі машинного навчання, керівник роботи Мілявський Юрій Леонідович, доцент кафедри ММСА.

затверджені наказом по університету від «___»_____ 20__ р. №_____

2. Термін подання студентом роботи _____

3. Вихідні дані до роботи _____
_____
_____
_____

4. Зміст роботи _____
_____
_____
_____

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо) _____
_____
_____
_____

6. Консультанти розділів роботи*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка

Студент _____ Скидан Б.О
(підпис) (ініціали, прізвище)

Керівник роботи _____ Мілявський Ю.Л.
(підпис) (ініціали, прізвище)

_____

РЕФЕРАТ

Дипломна робота: 47 с., 5 рис., 7 табл., 15 джерел.

ОЦІНКА ПЛАТОСПРОМОЖНОСТІ , НЕЙРОННІ МЕРЕЖІ, МАШИННЕ НАВЧАННЯ, МЕТОД К НАЙБЛИЖЧИХ СУСІДІВ, ГЛИБИННЕ НАВЧАННЯ

Об’єктом дослідження є оцінка платоспроможності позичальника у сучасному сенсі цього поняття.

Предметом дослідження є модель машинного навчання для прогнозування ймовірності дефолту позичальника.

Метою даної роботи є розробка методики для автоматизації оцінювання платоспроможності, шляхом оцінки історії банківських транзакцій.

Завданням цієї роботи є:

1. Розгляд існуючих на сьогоднішній день моделей для оцінки платоспроможності.
2. Дослідити загальні підходи щодо побудови моделей оцінки, використовуючи принципи та технології машинного навчання.
3. Провести аналіз вхідних даних, попередньо провести їх обробку, прибрати шум, та розробити моделі машинного навчання.
4. Виконати навчання моделі на тестових вхідних даних та подальше коригування моделі для отримання найкращих результатів.

ABSTRACT

Thesis contains 47 c., 5 Fig., 7 Tabl., 15 sources.

SOLVENCY FORECAST, NEURAL NETWORK, MACHINE LEARNING, K NEAREST NEIGHBORS METHOD, DEEP LEARNING

The object of this work is reviewing of existing models for forecasting solvency today.

The subject of my research is machine learning model for forecast prediction of client default.

The purpose of this work is development of a technique for automation of solvency estimation using history of bank transactions.

Consider common approaches of evaluation models using principles and technologies of machine learning.

Task of this work are:

1. To considerate models, for solvency assessment, that exists today.

2.To research general approaches of assessment model creation, using the principles and technologies of machine learning.

3. To analyze input data, previously make them suitable for researching and remove noise and to develop models of machine learning.

4. To train model on test data with next correcting model, for obtaining the best results.

Зміст

ВСТУП

Оцінка платоспроможності є важливою задачею для багатьох фінансових установ, зокрема для банків. Важливим етапом перевірки платоспроможності є оцінка кредитної історії клієнта. Якісна оцінка цієї історії є однією з складових успіху для банку. Потрібно знайти оптимальний спосіб оцінки платоспроможності, або відхилення від найкращого варіанту, коли кредити надаються, або занадто великій кількості людей, або навпаки занадто незначній, може означати різницю між банкрутством і рентабельністю.

Дивлячись на сучасні робочі умови, коли ситуація на ринку є досить непередбачуваною, також за умов постійних політичних потрясінь, фінансовим установам потрібно змінювати підхід для даних видів оцінок, виключивши людський фактор, задля отримання максимальної ефективності.

Оскільки оцінка платоспроможності фізичних осіб, задля винесення рішення по видачі кредиту, є процесом який можливо оптимізувати та автоматизувати. Тому це є актуальною задачею для фінансових установ.

Зважаючи на розвиток технологій в останні десятиліття, було б нерозумно використовувати застарілі методи та технології в роботі установ-кредиторів.

Метою даної роботи є розробка методики для автоматизації оцінювання платоспроможності, шляхом оцінки історії банківських транзакцій. Використання машинного навчання дозволяє забезпечити раціональну швидкість та якість розв’язання задачі.

Завданням цієї роботи є:

1. Розгляд існуючих на сьогоднішній день моделей для оцінки платоспроможності.
2. Розглянути загальні підходи щодо побудови моделей оцінки, використовуючи принципи та технології машинного навчання.

3. Провести аналіз вхідних даних, попередньо провести їх обробку, прибрати шум, та розробити моделі машинного навчання.

4. Виконати навчання моделі на тестових вхідних даних та подальше коригування моделі для отримання найкращих результатів.

Дана робота складається з 4 розділів. В першому проводиться огляд історичних принципів та підходів для оцінки платоспроможності. В другому розділі проходить ознайомлення з математичними моделями машинного навчання, та коригування моделей для поставленої задачі. В третьому розділі виконується аналіз та обробка вхідних даних, створення та коригування системи, яка була побудована на моделі машинного навчання та власне навчання цієї системи. В четвертому розділі проводиться аналіз економічної частини.

РОЗДІЛ 1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Вступ

Поняття платоспроможності в сучасному сенсі тісно пов’язане з поняттям кредитування. Оскільки саме платоспроможність вирішувала, на яку суму та чи взагалі можливо видати кредит певній особі. Тому з давніх часів актуальним було питання оцінки платоспроможності, задля мінімізації ризиків при наданні кредитів та позик.

Але з іншого боку згідно з теорії портфельного аналізу (Modern portfolio theory), яку розробив американський вчений-економіст Гаррі Марковіц, збільшення прибутку з портфеля здійснюється разом з збільшенням ризиків, як показано на рисунку 2. Тобто при бажанні отримати найбільший прибуток за найменший час, значно збільшується і ризик втратити свої кошти [13].

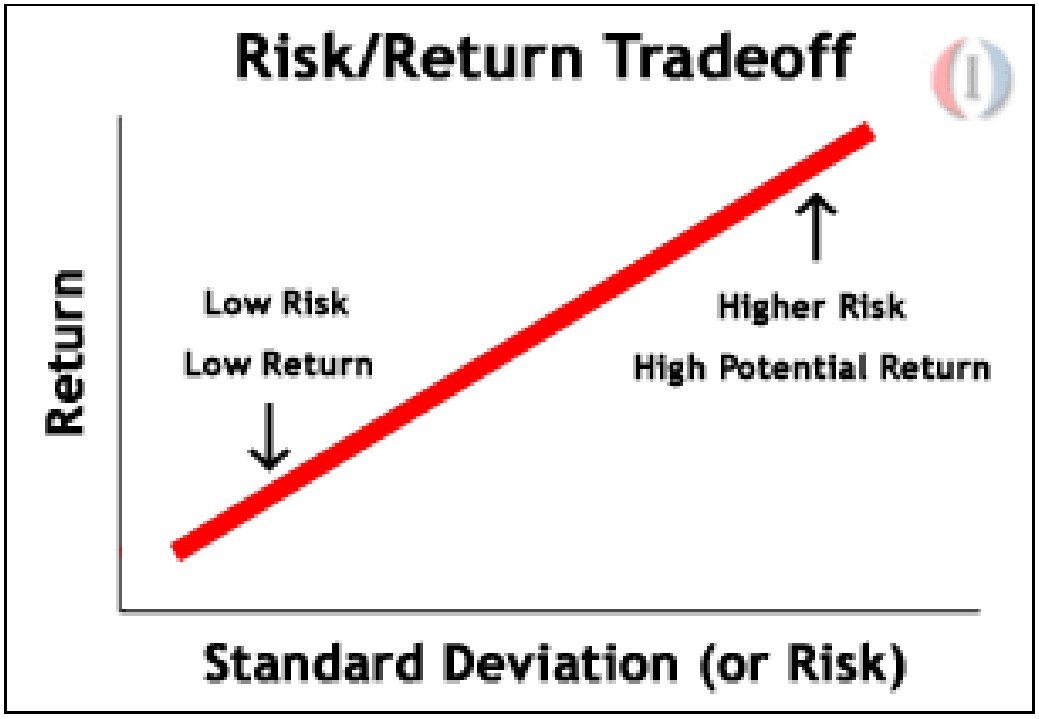

Рисунок 1 – Графік залежності ризику та доходу

Можна сказати що, тут працює українське прислів’я «Тихіше їдеш – далі будеш.» Тобто занадто ризикова політика фінансових установ загрожує в перспективі банкрутством. Якщо брати до уваги сучасний розвиток

технологій, фінансовим установам просто необхідно користуватись автоматизованими системами для оцінки платоспроможності, що можуть дозволити швидше та надійніше виконати роботу.

1.2 Поняття платоспроможності

Згідно з постановою Правління НБУ № 279 від 06.07.2000 платоспроможність це здатність позичальника здійснювати розрахунки за всіма видами своїх зобов'язань. Простими словами платоспроможність означає чи є у позичальника грошові кошти та їх еквіваленти в достатньому обсязі для розрахунків за кредиторською заборгованістю, яка потребує негайного погашення. Але існують деякі проблеми з подібним визначенням платоспроможності. Пов'язано це з тим що у багатьох позичальників, чи то підприємство чи то фізична особа є свої цикли виробництва-збуту і наприклад те що сьогодні у позичальника не вистачає коштів на погашення зобов'язань, не завжди означає що цих коштів не буде найближчим часом[1].

Поняття платоспроможності зазвичай використовується в контексті банківської справи та кредитування. Також крім вказаних в прямому визначенні грошових коштів та їх еквівалентів в оцінках платоспроможності в фінансових установах фігурує елемент довіри. Тобто двом людям з об'єктивно однаковим фінансовим положенням можуть дати різний висновок по кредиту.

Щодо ризику, сторони мають розуміти що завжди існує ймовірність того що ситуація локальна і глобальна може змінитися, можуть з'явитись різного роду проблеми, тому банки зазвичай перестраховуються від такого, додаючи відсотки по кредитам або потребують наявність поручителів чи застави.

1.3 Оцінка платоспроможності

В світі використовується декілька методик для оцінки платоспроможності. Для прикладу наведу дві найбільш відомі це методика PARSER та методика CAMPARI.

Розшифрування PARSER:

P – Person – інформація про особистість позичальника, його репутацію.

A – Amount – обґрунтування коштів, які просять.

R – Repayment – можливість повернення.

S – Security – забезпечення.

E – Expediency – доцільність.

R – Remuneration – банківський відсоток (винагорода банку).

Розшифрування CAMPARI:

C – Character – репутація позичальника.

A – Ability – оцінка бізнесу позичальника.

M – Means – аналіз необхідності звернення по позичку.

P – Purpose – з якою метою береться позичка.

A – Amount–сума кредиту (її обґрунтування).

R – Repayment – можливість повернення.

I – Insurance – страхування ризику [5].

Оцінка платоспроможності дає можливість не лише зробити аналіз фінансового стану підприємства та осіб, а й знайти негативні моменти в їх фінансових стратегіях та може допомогти передбачити майбутню стратегію підприємства. Це дає змогу вчасно вносити зміни до кредитних відносин підприємства та фінансових установ.

Такі установи повинні здійснювати оцінку фінансового стану позичальника перед наданням йому послуг кредитування , а в подальшому - щоквартально для визначення групи ризику та розміру відрахувань до загального і спеціального резервів згідно з Положенням про порядок формування резерву для відшкодування можливих втрат за позиками комерційних банків, яке затверджено Національним банком України.

В оцінці фінансового стану позичальника враховуються такі фактори як: обсяг реалізації, доходи та збитки, грошові потоки, рентабельність, фінансова стійкість, рентабельність, склад дебіторсько-кредиторської заборгованості, історія погашення заборгованостей та форма власності[14].

Залежно від стану платоспроможності, фінансової стійкості, «солідності» позичальника та його можливості виконувати боргові зобов'язання підприємства поділяються на 5 класів, які характеризують надійність позичальника:

1. Клас А – підприємство з дуже стійким фінансовим станом.
2. Клас Б - підприємство з стійким фінансовим станом.
3. Клас В – наявні ознаки фінансових проблем.
4. Клас Г – підприємство підвищеного ризику.
5. Клас Д – підприємство з незадовільним фінансовим станом.

На основі оцінки платоспроможності банк і ефективності комерційної угоди банк приймає рішення, щодо видачі підприємству кредиту та укладає з ним договір.

Зміст кредитного договору визначається підприємством і банком самостійно. В ньому передбачається ціль кредитування. умови і порядок видачі і погашення кредиту, спосіб забезпечення кредиту, процентні ставки за кредит, права і відповідальність сторін, інші умови.

Оцінювання платоспроможності або кредитне оцінювання було вперше застосовано в 60-х роках XX століття, для розуміння того чи зможуть позичальники повернути банку наданий їм кредит, чи будуть робити так як цього потребує процедура кредитування. Тоді під поняттям оцінювання платоспроможності розуміли лише колективне обговорення та винесення рішення «Прийнято/Відхилено» і багато людей і досі це так розуміють.

Закордоном діють різноманітні компанії, які розробляють моделі оцінки платоспроможності позичальників та пропонують свої послуги щодо аналізу за методиками, власної розробки.

При проведенні аналізу платоспроможності українські банківські установи керуються Положенням НБУ "Про порядок формування та використання резерву для відшкодування можливих втрат за кредитними операціями банків", затвердженим постановою Правління НБУ № 279 від 06.07.2000 р [1].

Окрім визначених у ньому обов'язкових вимог до оцінювання фінансового стану позичальників, банківські установи можуть використовувати інші методики, які регламентовані окремими внутрішніми банківськими положеннями щодо здійснення банківського кредитування та аналізу кредитоспроможності позичальників.

В літературі існують різні підходи щодо визначення видів платоспроможності в залежності від різних ознак, наведених на рисунку 2.

Рисунок 2 – Види платоспроможності.

Залежно від характеру та інформаційної бази дослідження стан платоспроможності може бути оцінений в статиці (статична платоспроможність) та в динаміці (динамічна платоспроможність).

Статична платоспроможність досліджується у певному часовому періоді й показує здатність підприємства до виконання планових платежів та термінових зобов'язань протягом періоду. Визначення її стану передбачає оцінку та порівняння розмірів грошових потоків (вхідних та вихідних). Інформаційною основою оцінки статичної платоспроможності є баланс

підприємств, в якому фіксується стан активів та пасивів підприємства. Оцінка динамічної платоспроможності проводиться на основі аналізу грошового обігу підприємства.

Залежно від терміну оцінювання існують поточна та перспективна платоспроможності.

Поточна платоспроможність характеризує платіжні можливості на дату або протягом періоду оцінки, перспективна визначає потенційні можливості виконання платіжних зобов'язань і витрат.

Залежно від характеру визначення обсягу платіжних засобів підприємства можна виділити такі види платоспроможності, як:

– грошова – за рахунок наявних коштів;

– розрахункова – за рахунок наявних грошових коштів та можливих (реальних) джерел їх зростання;

– майнова (потенційна) – за рахунок усіх видів оборотних активів підприємства (у разі їх умовного продажу);

Залежно від характеру визначення необхідного обсягу витрат грошових коштів платоспроможність може використовуватися для оцінювання можливості продовження фінансування поточної діяльності – фактична платоспроможність; ступеня покриття зовнішніх термінових зобов'язань та планових витрат загальна платоспроможність.

1.4 Історія розвитку

Сучасне оцінювання платоспроможності стало можливим з розвитком сучасного кредитування. Сучасне кредитування набуло свого виду з введенням від держав централізованого контролю над кредитними відносинами. Поява перших загальнонаціональних інститутів, наділених монопольними функціями по координації та нормативно-методичному забезпеченню кредитно-грошових відносин, сприяло формуванню повноцінної системи безготівкового грошового кругообігу, а також істотному розширенню

множини послуг і операцій комерційних банків, наприклад послуги по обслуговуванню фондових ринків.

В подальшому діяльність центральних банків розвивалась в направленні, насамперед використання кредитних важелів в якості одного з найбільш ефективних регуляторів ринкової економіки, що, потребувало певного посилення контролю з їх сторони за роботою не державних кредитних організацій. Нарешті, розвиток глобальних інформаційних технологій в економіці, формування глобальних банківських мереж, комп'ютерних комунікацій і баз даних дозволили вивести кредитні відносини на принципово новий якісний рівень, як в техніці обслуговування клієнта, так і поширенню їх на всі сфери фінансової діяльності, в тому числі – на міжнародних ринках.

Аж до другої світової війни комерційні банки розвинутих капіталістичних країн майже не надавали населенню грошові позички на споживчої цілі. Першими вступили на цей шлях комерційні банки США. Ще в 1920-1930 рр. група з кількох банків, під керівництвом одного з попередників нью-йоркських "City Corp" і "Bank of America", створили в себе відділи споживчого кредиту. Спочатку ця банківська група надавала позики на такі цілі, як оплата медичної допомоги, стоматологічних послуг, навчання тощо., але потім приступила і до видачі позичок на покупку в розстрочку споживчих товарів.

Після закінчення війни сектор споживчого кредиту став одним з найбільш швидкозростаючих сегментів ринку кредитних послуг комерційних банків. В інших західних країнах бум в області банківського кредитування споживчих потреб населення почався наприкінці 50-х років.

Таким чином, особливий розвиток споживчий кредит отримав в умовах загальної кризи капіталізму (головним чином після 2-ої світової війни 1939-1945) у зв'язку з різким посиленням невідповідності між зростанням виробництва і обмеженню попиту простих громадян.

Сьогодні ж банківський споживчий кредит отримав широке розповсюдження у всіх економічно розвинутих країнах.

Сучасні методи кредитування:

1. Суть першого методу полягає у тому, що питання про надання позички вирішується кожен раз в індивідуальному порядку. Позичка видається на задоволення певної цільової потреби в коштах. Цей метод застосовується при наданні позичок на певний термін, тобто термінових позик;

2. В другому методі позики надаються в межах раніше встановленого банків для позичальника ліміту кредитування, який використовується їм по мірі потреби шляхом оплати платіжних документів, що пред'являються до нього в межах певного періоду.

Така форма надання кредиту називається відкриттям кредитної лінії. Відкрита кредитна лінія приймає до оплати за рахунок кредиту будь-які розрахунково-грошові документи, передбачені кредитним договором, який приймається клієнтом і банком.

Розрізняють поновлювану і непоновлювану кредитні лінії. У випадку відкриття не поновлюваної кредитної лінії після видачі позики і її погашення відносини між банком і клієнтом закінчуються. При поновлюваній кредитній лінії кредит надається і погашається в межах встановленого ліміту заборгованості автоматично. Кредитна лінія може бути також цільовою, якщо вона відкривається для оплати поставок певних товарів в межах одного контракту, який реалізується протягом одного року або іншого періоду.

1.5 Висновки до розділу 1

В цьому розділі було розглянуто поняття платоспроможності, наведено точне визначення, згідно чинного законодавством України.

Було розглянуто не лише безпосередньо платоспроможність, а й суміжне з нею поняття кредитування та кредитної оцінки, як невід’ємну частину поняття платоспроможності. Наведено різни методики міжнародної

оцінки платоспроможності, такі як PARSER та CAMPARI. Показано класифікацію підприємств залежно від оцінки їх платоспроможності. Розглянуто приклади методики оцінювання платоспроможності закордоном.

Також, було наведено стислу історію оцінювання платоспроможності, як невід'ємну частину кредитування. Показано основні моменти цього процесу.

Як висновок можна сказати що, проведення оцінювання платоспроможності є одним з найважливіших процесів в банківській сфері, від якого залежить чи зможе банк отримувати прибуток та величину цього прибутку. Саме тому потрібно оптимізувати цей процес використовуючи сучасні технології, а саме машинне навчання.[]

РОЗДІЛ 2 ІСНУЮЧІ МОДЕЛІ ТА ПІДХОДИ ДО ОЦІНЮВАННЯ ПЛАТОСПРОМОЖНОСТІ.

Існують багато способів і моделей машинного навчання для автоматизації оцінювання платоспроможності.

Машинне навчання, як один з найуспішніших методів штучного інтелекту, здобуло великий успіх і популярність у багатьох сферах використання, таких як аналіз зображень, розпізнавання мови та аналіз тексту. Воно використовує контрольовані та неконтрольовані стратегії для навчання багаторівневих систем та методів в ієрархічній архітектурі для задач класифікації і розпізнавання образів. Недавні дослідження в сфері сенсорних мереж і технологій зв'язку дозволили створити колекцію (сховище) великих даних(big data). Також це сховище відкриває великі можливості для поширення e-commerce, управління виробничими процесами та технологічної медицини, що відкриває багато складних проблем для аналізу даних. Причинами цих проблем є: великий об'єм та різноманітність даних, потреба у великих швидкостях обробки та у достовірності результату.

За декілька останніх років глибинне навчання зіграло ключову роль у розв'язанні проблем пов`язаних великими обсягами даних. Нижче буде наведено короткий опис найпопулярніших моделей машинного навчання, які використовуються для вивчення особливостей аналізу великих даних.

2.1 Вступ

Великі дані (Big data) можна охарактеризувати за 4 характеристиками (так звані 4V) – великий об'єм(volume), велика варіативність(variety) , велика швидкість опрацювання(velocity) та велика достовірність (veracity).

Потреба у таких даних стає сьогодні дуже важливою. Наприклад Flickr генерує за день 3.6 ТБ даних, а гугл опрацьовує 20 000 ТБ даних щодня.

Національне агентство по безпеці повідомляє що приблизно 1.8 ПБ (10^{15} байт) поширюються щодня через інтернет. Ці цифри показують про великі об’єми даних. Також лише 20% з цих даних є структуровані (текст, зображення, відео, графіки та інше). Але більше 75% цих даних є неструктуровані. Велика швидкість опрацювання пояснює те що ці дані генеруються швидко та потребують опрацювання та аналізу у реальному часі (для різних онлайн сервісів, статистики і т.д.) Потреба у великій достовірності пояснюється тим що багато з цих даних являє собою шум, тобто, незакінчені, помилкові, неточні та надлишкові дані. Передбачається, що розмір цих даних збільшиться до 35 ЗБ(10^{21}) до 2020 року. Найпопулярнішими моделями глибинного навчання, яке є одним з найефективніших способів аналізу Big Data є:

1. Стековий автокодувальник.
2. Глибинна мережа переконань.
3. Згорткова нейронна мережа.
4. Рекурентна нейронна мережа.

2.2 Стековий автокодувальник.

Ця модель складається з декількох автокодувальників які зазвичай являють собою нейронну мережу прямого розповсюдження. Базові автоматичні кодувальники мають 2 фази. Фаза кодування(шифровки) і декодування (розшифровки).[15]

На фазі кодувальника поступаюча змінна x перетворюється на невідомий шар (невідомих шарів може бути більше ніж 1) h через функцію f.

$$h = f(W^{(1)}(x) + b^{(1)})$$

Далі невідома h повертається до значення y, яке буде використовуватися на стадії декодувальника.

$$y = g(W^{(2)}(h) + b^{(2)})$$

Зазвичай функції кодера та декодера є нелінійними відображеннями. Найбільш розповсюджені функції, показані на рисунку 1

Сігмоїд(Sigmoid):

$$f(x) = (1 + e(-x))^{-1}$$

гіперболічний тангенс(tanh):

$$f(x) = \frac{(e(x) - e(-x))}{(e(x) + e(-x))}$$

функція м'якого знаку (softsign) *:*

$$f(x) = \frac{x}{(1+|x|)}$$

та функція ReLU (Зглажена лінійна одиниця):

$$f(x) = \max(0,x)$$

На рисунку 3 показано графіки функцій кодувальника.

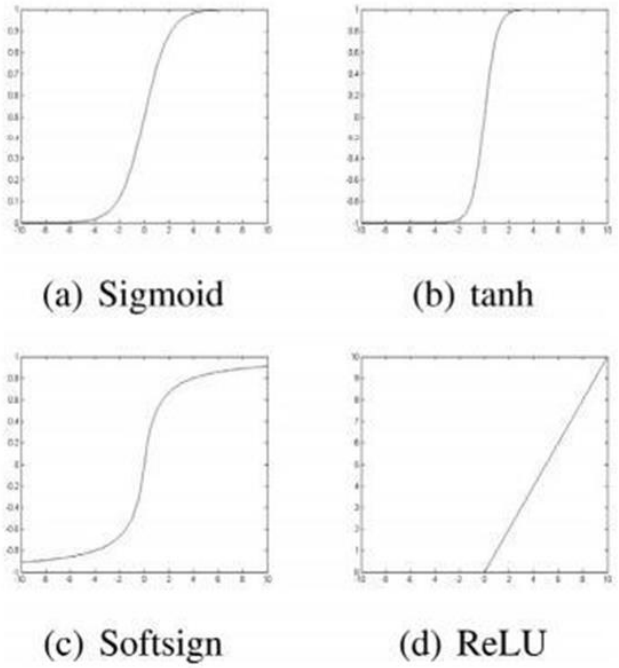

Рисунок 3 – Графіки функцій кодувальника.

Множина θ = { $W^{(1)}$, $W^{(2)}$, $b^{(1)}$, $b^{(2)}$ } – набір параметрів базового автоматичного кодувальника, яка зазвичай навчається шляхом мінімізації функції втрат J_θ відносно до m тренувальних прикладів.

$$J^\theta = \sum_{i=1}^{m} \left(y^i - x^i\right)^2$$

де x^i позначає *i* -й тренувальний приклад.

Базову модель кодувальника має декілька варіацій. Наприклад регуляризація, що допомагає значно зменшити ефект перенавчання. В цій варіації дещо модифікована функція втрат.[15]

$$J^\theta = \sum_{i=1}^{m} \left(y^i - x^i\right)^2 + \lambda \sum_{j=1}^{2} \left\| W^{(j)} \right\|$$

Ще одна модифікація називається розрідженим автоматичним кодувальником. Щоб автокодувальник став розрідженим, додають розріджену константу до прихованих модулів у відповідній функції втрат.

$$J^{\theta} = \sum_{i=1}^{m}\left(y^{i} - x^{i}\right)^{2} + \lambda \sum_{j=1}^{n} KL(p||p_{j})$$

де n показує кількість нейронів у прихованих шарах кодувальника, а інший показник показує відстань Кульбака-Лейбнера, яка розраховується (для j – ого нейрону) як

$$KL(p||p_{j}) = plog\left(\frac{p}{p_{j}}\right) + (1-p)\log(\frac{1-p}{1-p_{j}})$$

Де p показує заздалегідь визначений показник розрідженості, який близький до 0 і p_j показує середнє активаційне значення для j-ого нейрону у прихованому шарі кодувальника по всім навчальним прикладам.

Декілька таких автокодувальників будуть являти собою так званий стековий автокодувальник, який і буде являти нашу модель глибинного навчання.[15]

Навчання такої системи зазвичай складається зазвичай з 2 етапів – попереднє та фінальне навчання. На попередньому навчанні робиться безконтрольне проходження змінних від нижнього кодувальника до верхнього (як показано на рисунку 2)

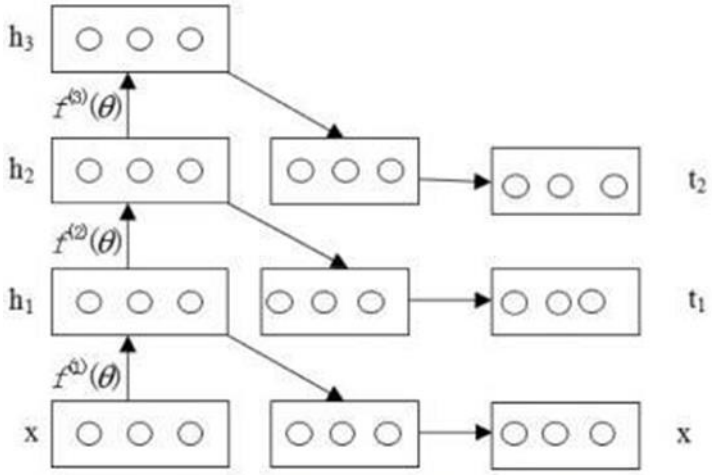

Fig. 3. Stacked auto-encoder for pre-training.

Рисунок 4 – Стековий автокодувальник.

Тобто проходячи авто кодера який приймає $h_0 = X$ на вхід і приймає Y_0 як вихід для навчання параметрів першого прихованого шару, потім h_1 робить все аналогічно для другого третього і прихованих шарів. Це повторюється до тих пір поки не будуть навчені всі приховані шари. В цьому і полягає етап попереднього навчання.

Згідно Хінтону, ця двоступенева стратегія навчання може уникати ефективних локальних оптимумів і досягати кращої збіжності в цій моделі глибокого навчання.

2.3 Глибинна мережа переконань.

Перша модель глибинного навчання, яка була успішно навчена це мережа переконань. Відмінності від попередньої це те що вона складається з декількох обмежених машин Больцмана, в яких прихований шар кожної підмережі слугує видимим шаром для наступної.

Типова машина Больцмана використовує алгоритм спрощення Gibbs'а для навчання параметрів. Особливо вони використовують умовну ймовірність $p(h,v)$ для обрахунку кожного значення параметру наступного прихованих шару. Потім використовує ту саму умовну ймовірність для обрахунку значень

параметрів вже видимих шарів. Цей процес повторюється до збіжності результатів [15].

Спільний розподіл обмежених машин Больцмана відносно параметрів визначається як

$$p(v,h,\theta) = \frac{\exp(-E(v,h,\theta))}{Z}$$

де $Z = \sum_v \sum_h \exp(-E(v,h,\theta)$ використано для нормалізації. E визначена як енергетична функція, що обчислюється через розподіл Бернуллі.

$$E(v,h,\theta) = -\sum_{j=1}^{I}\sum_{i=1}^{J} w_{ij} v_j h_i - -\sum_{i=1}^{I} b_i v_i - \sum_{j=1}^{J} a_j h_j$$

Де I та J це кількість видимих та прихованих параметрів.

$$\theta = (W, b, a)$$

множина параметрів обмеженої машини Больцмана.

Умовна ймовірність для кожного параметру машини з функцією Больцмана обраховується як:

$$p\left(v_j = 1 \middle| h, \theta\right) = f(\sum_{j=1}^{J} w_{ij} + b_i)$$

Також можливий варіант обмеженої машини Больцмана, де використовується розподіл Гауса-Бернуллі і енергетична функція обчислюється, як:

$$E(v,h,\theta) = -\sum_{i=1}^{I}\sum_{j=1}^{J} w_{ij} v_i h_j + \sum_{i=1}^{I}(v_i - b_i) - \sum_{j=1}^{J} a_j h_j$$

Відповідні умовні ймовірності для кожного видимого параметру обраховуються як:

$$p\left(v_j = 1 \middle| h,\theta\right) = N(\sum_{j=1}^{J} w_{ij} + b_i\,, 1)$$

Де v_j позначає дійсні значення, що задовольняють нормальному розподілу з мат. сподіванням $\sum_{j=1}^{J} w_{ij} h_j + b_i$ дисперсією

Обмежена машина Больцмана з таким розподілом може перетворювати дійснозначні випадкові змінні на бінарні.

Декілька таких машин являють собою глибинну мережу переконань. Тренування такої мережі проходить по аналогічній, як і в попередній моделі, двоступеневій системі. Тобто спочатку навчаємо початкові параметри жадібного шару допоки не отримаємо оптимального значення параметрів.

2.4 Згорткова нейронна мережа.

Ця мережа є найбільш широко розповсюджена серед мереж глибинного навчання для розпізнавання образів.

Шар згортки використовує операцію згортки для розподілу ваги, поки шар підвибірки використовуються для зменшення розмірності. Наприклад подається двовимірне зображення X, воно буде подано як послідовність = { x_1,x_2, . . . x_n} [13].

Шар згортки визначений як:

$$y_j = f(\sum_i K_{ij} \oplus x_i + b_j)\,,$$

де y_j визначає j вивід шару згортки і K_{ij} визначає ядро оператора згортки, $\oplus$ - операція згортки, b_j – зміщення. На додачу до цього f – нелінійна функція, зазвичай гіперболічний тангенс.

Підвибірковий шар націлений на зменшення розмірності вхідних даних. Як правило це здійснюється за допомогою операції вибіркового середнього або вибіркового максимуму. Потім результати роботи з'єднаних шарів передають на верхній шар для класифікації і розпізнання.

Для глибинного навчання така нейронна мережа має по декілька шарів згортки та шарів підвибірки для навчання.

В останні роки такі мережі досягли значного успіху в системах розпізнаванні мови.

2.5 Рекурентна нейронна мережа.

Моделі що були описані вище не беруть до уваги часові ряди. Така необхідність з’являється при розв’язанні задач розпізнавання рукописного тексту. [15]

Така мережа має декілька типів параметрів:

1. $\{x_1, x_2, \dots, x_t, x_{t+1} \dots\}$ – вхідні параметри.
2. $\{y_1, y_2, \dots, y_t, y_{t+1}, \dots\}$ – вихідні параметри.
3. $\{s_1, s_2, \dots, s_t, s_{t+1}, \dots\}$ – невідомі параметри.

На момент часу t рекурентна нейронна мережа опрацьовує x_t значення і попереднє приховане представлення s_{t-1} як введення для отримання нового представлення s_t, де f – функція кодувальника

$$s_t = f(x_t, s_{t-1})$$

Широковідома нейронна мережа є базовою, там представлення рахуються наступним чином:

$$s_t = f(W_{sx} + W_{ss}s_{t-1} + b_t)$$
$$y_t = g(W_{ys}s_t + b_y)$$

Де f та g функції шифрувальника та дешифрувальника відповідно і $\theta=\{W_{sx}, W_{ss}, b_s, W_{ys}, b_y\}$ - множина параметрів.

Рекурентна нейронна мережа показує залежність між поточним значенням x_t з попереднім x_{t-1} шляхом передачі попереднього прихованого представлення s_{t-1} у наступний момент часу. З теоретичної точки зору, рекурентна нейронна мережа може обробляти довільної довжини залежності. Однак ні, рекурентна нейронна мережа не підходить для обробки об’ємних залежностей через проблему градієнтного зникання з стратегією зворотного розповсюдження для навчання параметрів. Деякі моделі, такі як модель довгої короткочасної пам’яті, розв’язали цю проблему, шляхом запобігання зникання градієнту або градієнтного вибуху.

2.6 Метод K найближчих сусідів

kNN (k-Nearest Neighbors) - це алгоритм класифікації, проте це - лінивий класифікатор.[4]

Це означає, що в процесі навчання він не робить нічого, а тільки зберігає тренувальні дані. Він починає класифікацію тільки тоді, коли з'являються нові немарковані дані.

Активний же класифікатор створює класифікаційну модель в процесі навчання. Коли вводяться нові дані, такий класифікатор «згодовує» дані класифікаційної моделі.

Якими класифікаторами є C4.5, SVM і AdaBoost? На відміну від kNN, вони всі - активні класифікатори.

Ось чому:

C4.5 будує дерево рішень в процесі навчання;

SVM будує гіперплощину;

AdaBoost будує ансамблеву класифікацію.

kNN не будує ніякої класифікаційної моделі. Замість цього він просто зберігає розмічені тренувальні дані.

Коли з'являється нові нерозмічену дані, kNN проходить по 2 базовим крокам:

1.Спочатку він шукає k найближчих розмічених точок даних - іншими словами, k найближчих сусідів.

2.Потім, використовуючи класи сусідів, kNN вирішує, як краще класифікувати нові дані.

Як kNN розуміє, які точки знаходяться найближче? Для неперервних даних kNN використовує дистанційну метрику, наприклад, Євклідову дистанцію (метрику). Вибір метрики залежить від типу даних. Деякі радять навіть вибирати дистанційну метрику на підставі тренувальних даних. Є дуже багато нюансів , описаних у багатьох роботах по дистанційним метрик kNN.

При роботі з дискретними даними, вони спочатку перетворюються в неперервні. Ось 2 приклади:

Як kNN класифікує нові дані, якщо сусіди «не згодні»? kNN легко вирішує, до якого класу віднести дані, якщо всі сусіди належать одному класу. Логіка проста - якщо всі сусіди «згодні», то нові дані відводяться в їх клас.

Як kNN вирішує, до якого класу віднести дані, якщо сусіди не належать одному класу?

Для вирішення цієї проблеми використовуються 2 класичні техніки:

1. Прийняти за правильне рішення більшість «голосів». До якого класу належить найбільш кількість сусідів, туди і визначають точку даних.

2. Виконати те ж саме, але дати найближчих сусідів більшої ваги. Найпростіший спосіб зробити це - використовувати квантиль відстані. Якщо сусід відстоїть на 5 одиниць, то його вага буде 1/5. При збільшенні дистанції вага стає все менше і менше. Це якраз те, що нам потрібно.

Чи потребує цей метод навчання або він самонавчальний? Цей метод вимагає навчання, оскільки для kNN необхідний розмічений набір даних.

kNN легкий в розумінні і легко реалізуємий - це дві головні причини. Залежно від вибору дистанційній метрики, kNN може показувати досить точні результати.

Ось 5 речей, за якими потрібно стежити:

1. kNN може бути дуже ресурсоємким, якщо намагатися визначити найближчих сусідів на великому наборі даних.

2. Зашумлені дані можуть зіпсувати kNN-класифікацію.

3. Потрібно враховувати кількість значень. Характеристики з великою кількістю значень можуть впливати на дистанційну метрику, по відношенню до характеристик з меншою кількістю значень.

4. Оскільки обробка даних «відкладається», kNN зазвичай вимагає більше пам’яті, ніж активні класифікатори.

5. Вибір правильної дистанційній метрики дуже важливий для точності kNN.

2.7 Алгоритм C4.5

Алгоритм C4.5 будує класифікатор в формі дерева рішень. Щоб зробити це, йому потрібно передати набір вже класифікованих даних.

Як виглядає приклад використання алгоритму? Припустимо, що у нас є набір даних - це дані про групу пацієнтів. Ми знаємо різні параметри

кожного пацієнта: вік, пульс, кров'яний тиск, максимальне споживання кисню, історію сім'ї і так далі. Ці параметри називаються атрибутами.

Тепер на підставі цих атрибутів ми хочемо передбачити, чи може пацієнт захворіти на рак. Пацієнт може потрапити в один з 2 класів: буде хворіти на рак або не хворітиме на рак. Алгоритму C4.5 повідомляють клас кожного пацієнта. [5]

Ось у чому суть:

Використовуючи набір атрибутів пацієнта і відповідний клас, C4.5 будує дерево рішень, здатне передбачити клас для нових пацієнтів на підставі їх атрибутів.

Класифікація методом дерева рішень створює якусь подобу блок-схеми для розподілу нових даних. Якщо повернутися до прикладу з пацієнтом, то гілка блок-схеми може виглядати так:

- у пацієнта в історії сім'ї є захворювання на рак;
- у пацієнта є ген, який присутній у пацієнтів, хворих на рак;
- у пацієнта пухлина;
- розмір пухлини більше 5 см.

Таким чином:

Цей метод вимагає навчання, тут тренувальний набір даних розмічається класами. Знову повертаючись до прикладу з пацієнтами, відзначимо, що C4.5 не вирішує самостійно, захворіє пацієнт раком чи ні. Як ми вже говорили, він створює дерево рішень, яке використовується для прийняття рішень.

Ось відмінності C4.5 від інших систем, що використовують дерева рішень:

По-перше, C4.5 використовує приплив інформації , при створенні дерева рішень.

По-друге, хоча інші системи також проріджують гілки дерева рішень, C4.5 використовує проріджування , щоб уникнути перенавчання. Відсікання гілок поліпшує модель.

По-третє, C4.5 може працювати з дискретними і безперервними значеннями. Він робить це, обмежуючи діапазони і встановлюючи пороги даних, звертаючи безперервні дані в дискретні.

Нарешті, пропущені дані обробляються своїми власними способами .

Приклад дерева рішень C4.5 наведено на рисунку

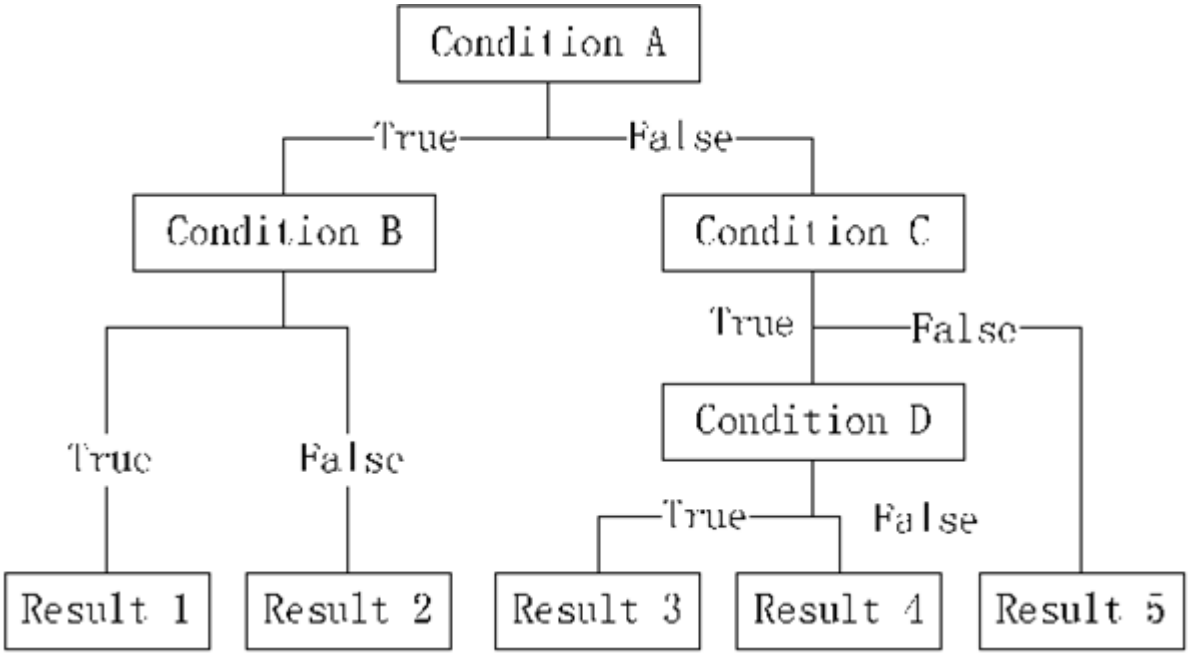

Рисунок 5 – Дерево рішень Алгоритму C4.5

2.8 Adaptive boosting

Бустінг (англ. Boosting) - мета-алгоритм машинного навчання . Основною ідеєю є комбінування слабких функцій, які будуються в ході ітеративного процесу, де на кожному кроці нова модель навчається з використанням даних про помилки попередніх. Сильний навчальний алгоритм є класифікатором, добре корелює з вірною класифікацією, на відміну від слабкого

Одним з недоліків бустінга є те, що він може приводити до побудови громіздких композицій, що складаються з сотень алгоритмів. Такі композиції виключають можливість змістовної інтерпретації, вимагають великих обсягів пам'яті для зберігання базових алгоритмів і істотних витрат часу на обчислення класифікацій.

Бустінг являє собою композицію алгоритмів, в яких помилки окремих алгоритмів взаємно компенсуються. Наприклад, в задачі класифікації на два класи Y={−1,+1} в якості простору оцінок приймають R=R и C(b(x))=sign(b(x)). Тоді базові алгоритми повертають {-1, 0, 1}. 0 означає що алгоритм відмовляється від класифікації об'єкта (не визначено) і b(x) не враховується в подальшій композиції. Отримаємо шукану композицію:

$$a(x) = C(F(b_1(x), \ldots, b_T(x))) = \text{sign}(\sum_{t=1}^{T} \alpha_t b_t(x)), x \in X$$

Велика частина алгоритмів бустінга ґрунтується на ітеративному навчанні слабких класифікаторів з подальшою збіркою їх у сильний класифікатор. Коли вони додаються, їм зазвичай приписуються ваги, зазвичай пов'язані з точністю навчання . Після додавання слабкого класифікатора, ваги перераховуються («перерахунок вагових коефіцієнтів»). Невірно класифіковані вхідні дані отримують більшу вагу, а правильно класифіковані екземпляри втрачають вагу. Таким чином, подальше слабке навчання фокусується на прикладах, де попередні слабкі навчання дали помилкову класифікацію. [7]

Алгоритм:

Дано: X^l– навчальна вибірка;

b_1 ,…, b_t– базові алгоритми класифікації;

Ініціалізація ваг об'єктів: $p_i = \frac{1}{l}, i = \frac{1}{l}, \ldots, l$;

Для всіх t=1,…, T, поки не виконується критерій зупинки.

Знаходим класифікатор $b_t: X \to \{-1, +1\}$ який мінімізує зважену помилку класифікації;

$$b_t = argmin_b\, Q(b, W^l);$$

Перераховуємо коефіцієнт зваженого голосування для алгоритму класифікації

$$b_t: \alpha_t = \frac{1}{2} \ln \frac{1 - Q(b, W^l)}{Q(b, W^l)};$$

Перерахунок ваг об’єктів:

$$w_i = w_i \, exp\left(-\alpha_t \, y_i \, b_t(x_i)\right), i = 1, \dots, l;$$

Нормування ваг об’єктів

$$w_0 = sign(\sum_{j=1}^{l} w_j; \; w_i = \frac{w_i}{w_0} \; i = 1, \dots, l;$$

Повертаємо

$$a(x) = sign\left(\sum_{i=1}^{T} \alpha_i \, b_i(x)\right)$$

Переваги алгоритму:

1. Хороша узагальнююча здатність. В реальних задачах (не завжди, але часто) вдається будувати композиції, що перевершують за якістю базові алгоритми. Узагальнююча здатність може поліпшуватися (в деяких завданнях) у міру збільшення числа базових алгоритмів.

2. Простота реалізації. Власні накладні витрати бустінга невеликі. Час побудови композиції практично повністю визначається часом навчання базових алгоритмів.

3. Можливість ідентифікувати об'єкти, які є шумовими викидами.

Недоліки алгоритму:

AdaBoost схильний до перенавчання при наявності значного рівня шуму в даних. Експоненціальна функція втрат занадто сильно збільшує ваги найбільш важких об'єктів, на яких помиляються багато базові алгоритми.

Однак саме ці об'єкти найчастіше виявляються шумовими викидами. В результаті AdaBoost починає налаштовуватися на шум, що веде до перенавчання. Проблема вирішується шляхом видалення викидів або застосування менш агресивних функцій втрат.

AdaBoost вимагає досить ємких навчальних вибірок. Жадібна стратегія послідовного додавання призводить до побудови не оптимальні набору базових алгоритмів. Для поліпшення композиції можна періодично повертатися до раніше побудованим алгоритмам і навчати їх заново. Для поліпшення коефіцієнтів можна оптимізувати їх ще раз після закінчення процесу бустінга за допомогою якого-небудь стандартного методу побудови лінійної розділяє поверхні. Рекомендується використовувати для цієї мети SVM (машини опорних векторів).

Бустінг може призводити до побудови громіздких композицій, що складаються з сотень алгоритмів. Такі композиції виключають можливість змістовної інтерпретації, вимагають великих обсягів пам'яті для зберігання базових алгоритмів і істотних витрат часу на обчислення класифікацій.

2.9 Висновок до розділу 2

В цьому розділі було розглянуто основні технології машинного навчання. Ретельно вивчено і показано різні методики для створення систем машинного навчання, завдяки яким буде функціонувати автоматизація оцінювання платоспроможності.

РОЗДІЛ 3 ПОБУДОВА МОДЕЛІ ПРОГНОЗУВАННЯ ПЛАТОСПРОМОЖНОСТІ

3.1 Вступ

Основною метою цієї роботи є побудова моделі, яку різні фінансові установи можуть використовувати для прогнозування платоспроможності своїх клієнтів. Для її досягнення будемо використовувати KDD(Knowledge discovery in databases), або виявлення знань у базах даних.

Об'єднуючою метою процесу KDD є отримання знань з даних у контексті великих баз даних.

Це робиться за допомогою методів (алгоритмів) інтелектуального аналізу даних для вилучення (ідентифікації) того, що вважається знанням, відповідно до специфікацій заходів і порогових значень, використовуючи базу даних разом з будь-якою необхідною попередньою обробкою та перетвореннями цієї бази даних.

Основні етапи цієї методики:

- Усвідомлення:
  - Структури знань
  - Цілей користувача
- Створення цільового набору даних: вибір набору даних або фокусування на підмножині змінних, або зразках даних, на яких має виконуватися виявлення знань.
- Очищення та попередня обробка даних:
  - Видалення шуму або викидів
  - Збір необхідної інформації для моделювання
- Звуження даних:
  - Пошук корисних функцій для представлення даних залежно від поставлених цілей.

- Використання методів зменшення розмірності або перетворення для зменшення ефективного числа розглянутих змінних або пошуку інваріантних представлень даних.

– Видобуток даних.

– Пошук фіч в певній репрезентативній формі за допомогою таких алгоритмів та класифікаційних правил як дерева рішень, авторегресія, алгоритми кластеризації, тощо.

– Інтерпретація видобувних паттернів.

– Об'єднання виявлених знань.

Узагальнено ці етапи показано на рисунку 5.

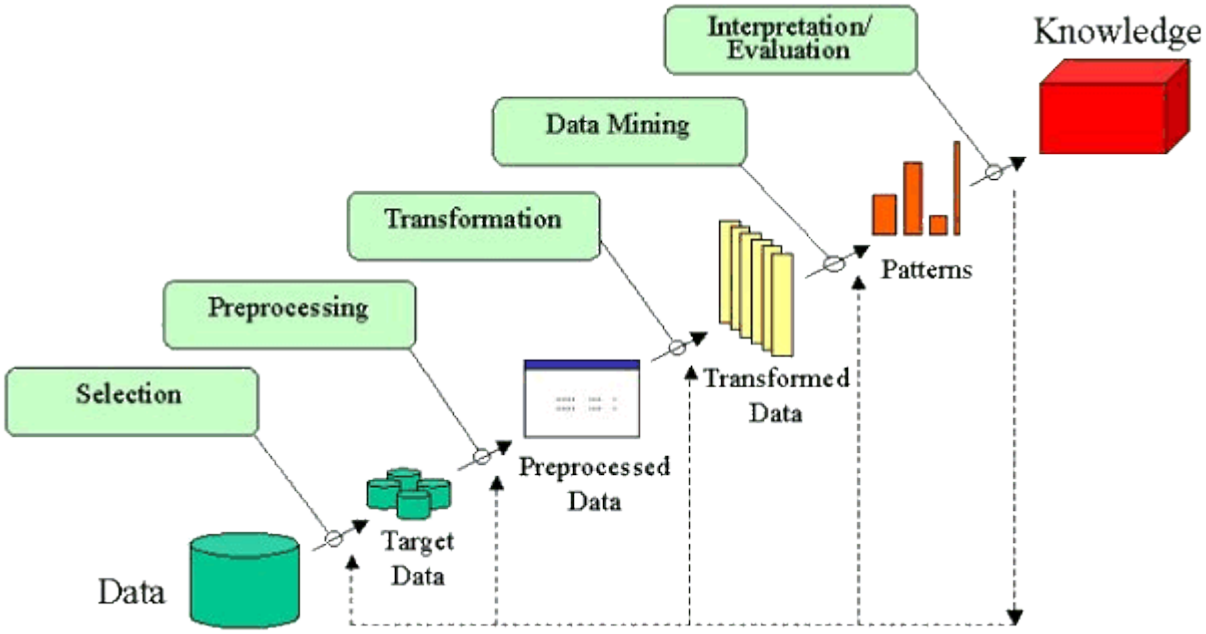

Рисунок 5 – Основні етапи KDD

3.2 Опис вхідних даних

Маємо на вхід базу даних банківських транзакцій, що має 6 змінних, а саме:

– advanceID – ідентифікатор транзакції;

– LoanNum – порядковий номер кредиту клієнта;

– ProductName – тип кредиту, може приймати такі значення:

- SACC - короткостроковий кредит на малу суму;
- MACC - довгостроковий кредит на більшу суму;
- Loan – те ж саме що і SACC, але трохи відрізняються умови кредитування;
- AGDateCreated – дата і час транзакції
- Fails35NoPayIn90 – чи були затримки в виплатах на 35 днів протягом 90 днів
- LoanWriteOff – чи списаний кредит.

На рисунку 6 показано результати статистичного аналізу вхідних даних.

	LoanNum	Fails35NoPayIn90	LoanWriteOff
count	1.057167e+06	1.055520e+06	1.057167e+06
mean	7.561337e+00	5.362381e-02	5.631277e-02
std	9.677870e+00	2.252739e-01	2.305248e-01
min	1.000000e+00	0.000000e+00	0.000000e+00
25%	2.000000e+00	0.000000e+00	0.000000e+00
50%	4.000000e+00	0.000000e+00	0.000000e+00
75%	1.000000e+01	0.000000e+00	0.000000e+00
max	1.450000e+02	1.000000e+00	1.000000e+00

Рисунок 6 – Статистичні дані.

Тут показано лише ті поля які піддаються статистичній обробці (числові змінні). З них можна побачити що кількість записів - 1056167. Середні значення, середньоквадратичні відхилення, мінімальні та максимальні значення LoanNum, Fails35NoPayIn90 та LoanWriteOff

3.3 Попередня обробка даних

Для кращого моделювання, дані потрібно попередньо обробити. Маємо базу даних з 6 полями:

1. advanceID
2. LoanNum
3. ProductName

4. AGDateCreated
5. Fails35NoPayIn90
6. LoanWriteOff

В вигляді поля для передбачення візьмемо LoanWriteOff – що показує чи виплачено кредит. AdvanceID – це лише унікальний ідентифікатор транзакції, що не представляє для нас ніякої цінності, тому його можемо відкинути. В полі ProductName, що містить лише певний перелік типів кредитів, замінимо кожен тип на натуральне число, а саме:

1.SACC замінимо на 1.

2. MACC замінимо на 2.

3. Loan замінимо на 3.

Оскільки дані з самого початку були подані в правильному порядку, відносно дати і часу транзакцій, то AGDateCreated також не буде впливати на результати моделювання, тому її можна викинути.

В результаті попередніх дій отримаємо таблицю з 4 полів та 1057168 записів. Їх опис наведено на рисунку 7.

	LoanNum	ProductName	Fails35NoPayIn90	LoanWriteOff
count	1.057167e+06	1.057167e+06	1.055520e+06	1.057167e+06
mean	7.561337e+00	2.222878e+00	5.362381e-02	5.631277e-02
std	9.677870e+00	9.651487e-01	2.252739e-01	2.305248e-01
min	1.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00
25%	2.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00
50%	4.000000e+00	3.000000e+00	0.000000e+00	0.000000e+00
75%	1.000000e+01	3.000000e+00	0.000000e+00	0.000000e+00
max	1.450000e+02	3.000000e+00	1.000000e+00	1.000000e+00

Рисунок 7 – Статистичний опис змінних після обробки.

3.4 Опис моделювання

Для моделювання було обрано мову програмування Python 3.7 та бібліотеку sklearn. Також в моделюванні використовувались, такі бібліотеки Pyhton,як numpy,pandas та scipy.

Для навчання моделі, множину вхідних даних було розділено на 2 частини функцією train_test_split. Частина train складає 80% від початкової множини і слугує для тренування алгоритму. Частина test складає 20% і потрібна вона для перевірки якості алгоритму.

В моделюванні використовувався алгоритм K найближчих сусідів (kNN – k nearest neighbors) з кількістю сусідів – 6. Використовуючи такі налаштування алгоритму було отримано такий результат збіжності передбачення моделі та тестової множини на 0.9998675709677725, що по суті означає явне перенавчання алгоритму. Метрика що використовувалась метрика Мінковськи (Minkowski distance)

$$D(X,Y) = \left(\sum_{i=1}^{n} |x_i - y_i|^2\right)^{\frac{1}{2}}$$

де X $=\{x_i\}$, Y $=\{y_i\}$ i=1..n – вектори, між якими потрібно знайти відстань;

n - розмірність векторів;

D(X,Y) – відстань між векторами X та Y;

3.5 Висновки до розділу 3

В цьому розділі було проведено обробка даних, моделювання та аналіз результатів. Показано методику оцінку платоспроможності, маючи історію банківських транзакцій. Отриманий результат – майже достовірна точність(більше 99%).

ВИСНОВКИ

Метою даної роботи є дослідження в галузі оцінювання платоспроможності. В першому розділі було ознайомлення з теоретичними та історичними моментами. В другому розділі більше наукові моменти машинного навчання, що використовується для автоматизації оцінки платоспроможності. Було наведено такі алгоритми як

- Стековий автокодувальник
- Глибинна мережа переконань
- Згорткова нейронна мережа
- Рекурентна нейронна мережа
- Метод K найближчих сусідів
- Алгоритм C4.5
- Adaptive boosting

Показано їх недоліки та переваги.

Також у даній роботі були наведені методики оцінки платоспроможності з використанням алгоритму K найближчих сусідів на мові програмування Pyhton 3.7 та бібліотеки sklearn.

ПЕРЕЛІК ДЖЕРЕЛ

1. Про затвердження Положення про порядок формування і використання резерву для відшкодування можливих втрат за позиками комерційних банків. URL:https://zakon.rada.gov.ua/laws/show/v0471500-97 (дата звернення 31.03.2019)

2. Breiman, L. [2000] Some infinity theory for predictor ensembles, Technical Report 579, Statistics Dept. UCB 325–350

3. Amit, Y. & Geman, D. (1997). Shape quantization and recognition with randomized trees. *Neural Computation, 9,* 1545–1588.

4. Thomas G Dietterich, Ensemble Methods in Machine Learning URL: http://web.engr.oregonstate.edu/~tgd/publications/mcs-ensembles.pdf (дата звертання 01.05.2019)

5. Zhi-Hua Zhou, Ensemble Learning URL: https://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/springerEBR09.pdf (дата звертання 01.05.2019)

6. Documentation of scikit-learn 0.21.2 URL: https://scikit-learn.org/stable/documentation.html (дата звертання 02.05.2019)

7. Ensemble methods: bagging, boosting and stacking URL: https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205 (дата звертання 01.05.2019)

8. Ensemble Methods to Optimize Machine Learning Models URL: https://hub.packtpub.com/ensemble-methods-optimize-machine-learning-models/(дата звертання 01.05.2019)

9. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to Boosting. Journal of Computer and System Sciences 55(1) (1997) 119–139

10. Breiman, L.: Bagging predictors. Machine Learning 24(2) (1996) 123–140

11. Wolpert, D.H.: Stacked generalization. Neural Networks 5(2) (1992) 241–260

12. Schapire, R.E.: The strength of weak learnability. Machine Learning 5(2) (1990) 197–227

13. Markowitz, H.M. (1959). Portfolio Selection: Efficient Diversification of Investments. New York: John Wiley & Sons. (reprinted by Yale University Press, 1970

14. Литвин Б. М., Стельмах М. В. Фінансовий аналіз. Навчальний посібник. Київ: ХайТек Прес, 2008. 336 с

15. Qingchen Zhang, Laurence T.Yang, Zhikui Chen, Peng LI A survey on deep learning for big data (2017) URL: https://www.sciencedirect.com/science/article/pii/S1566253517305328?via%3Dihub. (дата звернення 31.05.2018)

ДОДАТОК А КОД ПРОГРАМНОГО ПРОДУКТУ

```
import numpy as np
import pandas as pd
import sklearn
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.utils.validation import assert_all_finite
from sklearn.metrics import roc_auc_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn import preprocessing
from sklearn.svm import SVR
sklearn.set_config(assume_finite=True)
def noNa(a):
    nans = np.isnan(a)
    infs = np.isinf(a)
    a[nans] = 0
    a[infs] = 0
    return a


dataset = pd.read_csv(r"C:\Users\Пользователь\PycharmProjects\Dipl\saveOnlyWithNumbers.csv", sep=',', index_col=0)
print(dataset)
dataset = dataset.fillna(dataset.median(axis=0))
print("l")
X = dataset.iloc[:, 0:4].values
y = dataset.iloc[:, 3].values

print(X)
print("_____")
print(y)
lab_enc = preprocessing.LabelEncoder()

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
print(X_train,X_test)
print("2")
classifier = RandomForestClassifier(n_estimators=10, random_state=0)
classifier = KNeighborsClassifier(n_neighbors=6)
classifier = KNeighborsClassifier(n_neighbors=6, leaf_size=400)
print("3")
classifier.fit(X_train, y_train)
print("3.1")

y_pred = classifier.predict(X_test)
print("4")
y_pred_train = classifier.predict(X_train)
print("5")
print("=========y_pred====================")
```

```
print(classifier.score(y_pred, y_test))
print("6")
print("==========y_pred_train====================")

print(classifier.score([y_pred_train], [y_train]))
print("7")
print(classifier.score([y_test], [y_pred]))
print("8")
 -*- coding: utf-8 -*-

from sklearn.datasets import load_iris
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score
import sklearn
import pandas as pd
import numpy as np
from scipy import stats

sklearn.set_config(assume_finite=True)


def noNa(a):
    nans = np.isnan(a)
    infs = np.isinf(a)
    a[nans] = 0
    a[infs] = 0
    return a


 ['advanceID',
 'LoanNum',
 'ProductName',
 'AGDateCreated',
 'Fails35NoPayIn90',
 'LoanWriteOff']
def mode(x):
    try:
        return stats.mode(x)[0][0]
    except:
        return np.nan


def mean(x):
    try:
        return np.mean(x)
    except:
        return np.nan


 dataset = pd.read_csv(r"C:\Users\Пользователь\PycharmProjects\Dipl\all.rpt", sep='\t')
```

```
 grouped = dataset.groupby('ProductName').agg(
    {"Fails35NoPayIn90": mean,"LoanWriteOff" : mean})
 trg = dataset[['Fails35NoPayIn90', 'LoanWriteOff']]
 trn = dataset[['LoanNum', 'ProductName', 'AGDateCreated']]
 Xtrn, Xtest, Ytrn, Ytest = train_test_split(trn, trg, test_size=0.2)
 TestModel = pd.DataFrame()
 tmp = {}
 for i in range(Ytrn.shape[1]):
     KNeighborsClassifier.fit(Xtrn, Ytrn[:, i])
     tmp['R2_Y%s' % str(i + 1)] = r2_score(Ytest[:, 0], KNeighborsClassifier.predict(Xtest))
     TestModel = TestModel.append([tmp])
print(grouped)
 print(dataset.head())
 print(dataset)
 print(dataset.loc[0, :])
 dataset1 = pd.read_csv(base_dir + "beh_all.rpt", error_bad_lines=False, sep='\t')
 print(dataset1.iloc[0, :])

 dataset2 = pd.read_excel("resources/fieldids.xlsx", error_bad_lines=False)
dataset2 = dataset.drop("advanceID",axis=1)
  dataset2.to_csv(r"C:\Users\Пользователь\PycharmProjects\Dipl\saveWithoutID.csv", index=False, sep=",")
dataset4 = dataset3.loc[:, ['advanceID', 'FieldId', 'Value', 'ChangeType', 'Timestamp', 'FocusTimestamp', 'AvgSpeed', 'SdSpeed', 'AuxText',
'IsMobileDevice', 'Browser', 'SourceType', 'CreateDate', 'PageId', 'DeviceId', 'Fails35NoPayIn90']]
 print("===============FILTERING FINISHED===============")

 dataset4.to_csv("/home/kaambaalaa/diploma/filtered.csv", index=False)
 print("===============FILTERING SAVED===============")

 print(dataset1.iloc[:, 0:4])
 print(dataset1.iloc[:, 0:4].values)
 X = dataset1.iloc[:, 0:4].values
 y = dataset1.iloc[:, 4].values

 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
 dataset = pd.read_csv(r"C:\Users\Пользователь\PycharmProjects\Dipl\saveWithoutID.csv", sep=',')
 dataset = dataset.replace('Loan*',1)
 dataset = dataset.replace('Loan',2)
 dataset = dataset.replace('SACC4%',3)
 dataset = dataset.replace('MACC',4)
 dataset = dataset.replace('SACC3%',5)
 print(dataset)
 dataset =pd.read_csv(r"C:\Users\Пользователь\PycharmProjects\Dipl\saveWithoutID.csv", sep=',')
 dataset.drop('AGDateCreated',axis=1)
 dataset = dataset.replace('Loan*',1)
 dataset = dataset.replace('Loan',2)
 dataset = dataset.replace('SACC4%',3)
 dataset = dataset.replace('MACC',4)
 dataset = dataset.replace('SACC3%',5)
 dataset.to_csv(r"C:\Users\Пользователь\PycharmProjects\Dipl\saveOnlyWithNumbers.csv", sep=',')
dataset = pd.read_csv(r"C:\Users\Пользователь\PycharmProjects\Dipl\saveOnlyWithNumbers.csv", sep=',', index_col=0)
dataset1 = dataset.groupby('LoanNum').agg({"LoanWriteOff": mean, "Fails35NoPayIn90": mean})
print(dataset1)
```

```
dataset1.to_csv(r"C:\Users\Пользователь\PycharmProjects\Dipl\saveLoanNums.csv", sep=",")
print("==============================")
import numpy as np
import pandas as pd
import sklearn
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.utils.validation import assert_all_finite
from sklearn.metrics import roc_auc_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn import preprocessing
from sklearn.svm import SVR

sklearn.set_config(assume_finite=True)
def noNa(a):
    nans = np.isnan(a)
    infs = np.isinf(a)
    a[nans] = 0
    a[infs] = 0
    return a
dataset = pd.read_csv(r"C:\Users\Пользователь\PycharmProjects\Dipl\saveOnlyWithNumbers.csv", sep=',', index_col=0)
print(dataset)
dataset = dataset.fillna(dataset.median(axis=0))
print("l")
X = dataset.iloc[:, 0:4].values
y = dataset.iloc[:, 3].values
print(X)
print("_____________________")
print(y)
lab_enc = preprocessing.LabelEncoder()
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
print(X_train,X_test)
print("2")
classifier = RandomForestClassifier(n_estimators=10, random_state=0)
classifier = KNeighborsClassifier(n_neighbors=6)
classifier = KNeighborsClassifier(n_neighbors=6, leaf_size=400)
print("3")
classifier.fit(X_train, y_train)
print("3.1")
y_pred = classifier.predict(X_test)
print("4")
y_pred_train = classifier.predict(X_train)
print("5")
print("=========y_pred====================")
print(classifier.score(y_pred, y_test))
print("6")
print("==========y_pred_train====================")
print(classifier.score([y_pred_train], [y_train]))
print("7")
print(classifier.score([y_test], [y_pred]))
print("8")
```